

Verteilte Systeme Lab 4

Sommersemester 2023

Hochschule Karlsruhe

Aufgabe 1.) Zustandsdiagramm

https://github.com/dnszlr/vislab/blob/master/lab4/lab4-base/attachments/state_diagram_transactions.png

Aufgabe 2.) Analyse der Client Requests

```
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/f605402d-ce8f-479f-8ced-ehdca5cf96db, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193144361, date=1687279484334, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/79a74ac3-f6b8-4dbd-8dc0-dac708275077, hotel=Interconti, expires=1687193204370, date=1687279484334, bookingDate=1687215600000, confirmed
## Request 5/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/3c469501-9a31-4781-8616-8104c6ff497d, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193144607, date=1687279484582, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/677885e4-c4fb-4243-8803-f3dea97468f5, hotel=Interconti, expires=1687193204610, date=1687279484582, bookingDate=1687215600000, confirmed
## Request 6/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/a525dfdf-edc5-4621-a6f1-02619c8185a9, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193144857, date=1687279484833, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/a016c9af-e9a9-45d8-90a2-34d92ea9acc, hotel=Interconti, expires=1687193204860, date=1687279484833, bookingDate=1687215600000, confirmed
## Request 7/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/fa819175-da1d-4a3f-811c-50e8aae89faf, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193145106, date=1687279485084, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/67d8c0dd-f08c-4584-9adf-eaf087443984, hotel=Interconti, expires=1687193205115, date=1687279485084, bookingDate=1687215600000, confirmed
## Request 8/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/27349ee0-d5ff-4e8d-bcaa-1efaae75a2f4, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193145353, date=1687279485332, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/5f9ba111-dae7-4702-a4d0-630261a0e074, hotel=Interconti, expires=1687193205363, date=1687279485332, bookingDate=1687215600000, confirmed
## Request 9/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/d3a73197-a075-447b-ac52-ace5d5d37bc, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193145589, date=1687279485568, bookin
Output from Server: HotelReservationDoc [name=Christian, url=http://localhost:4434/hotel/57c0daa6-987e-4aae-a779-515fa1770dc3, hotel=Interconti, expires=1687193205597, date=1687279485568, bookingDate=1687215600000, confirmed
## Request 10/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/d8def801-9a07-43eb-9a3d-b8b161877052, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193145822, date=1687279485802, bookin
Failed : HTTP error code : 409
Output from Server: HotelReservationDoc [name=Christian, url=null, hotel=Interconti, expires=0, date=1687279485802, bookingDate=0, confirmed=false]
## Request 11/11
Output from Server: FlightReservationDoc [name=Christian, url=http://localhost:4434/flight/9f77e8aa-52e2-478a-be12-63d89c6105a1, airline=airberlin, from=Karlsruhe, to=Berlin, expires=1687193146056, date=1687279486037, bookin
Failed : HTTP error code : 409
Output from Server: HotelReservationDoc [name=Christian, url=null, hotel=Interconti, expires=0, date=1687279486037, bookingDate=0, confirmed=false]
```

Tatsächlich scheitern die Anfragen ab der 10. Anfrage, da die maximale Anzahl der buchbaren Hotels pro Tag bei 9 liegt.

```
// Maximum number of bookings per day
1 usage
private static final int MAXBOOKINGS = 10;

// Duration for confirming a reservation
```

```
public HotelReservation createReservation(String name, String hotel, long date) throws BookingException {
    long normdate = DateUtil.normalize(date);
    int bookings = 1;

    lock.lock();
    try {
        for (Iterator<HotelReservation> iterator = reservations.values().iterator(); iterator.hasNext(); ) {
            HotelReservation res = (HotelReservation) iterator.next();
            if (res.getBookingDate() == normdate && res.hotel.equals(hotel)) {
                bookings ++ 1;
            }
        }

        if (bookings < MAXBOOKINGS) {
            HotelReservation reservation = new HotelReservation(name, hotel, date);
            reservations.put(reservation.id, reservation);

            return reservation;
        } else {
            throw new BookingException(BookingError.MAX_NUM_OF_BOOKINGS);
        }
    }
}
```

Aufgabe 3.) Diskussion ACID-Transaktionen

Atomicity:

Durch den Rollback im Falle eines Timeouts ist Atomicity gegeben, es sei denn es kommt in seltenen Fällen zu einer Komplikation und eine der beiden Reservierungen kann nicht bestätigt werden. In diesem Fall wäre das System in einem inkonsistenten Zustand, welcher nicht rückgängig gemacht werden kann.

Consistency:

Wenn der Fall eintritt, der bereits bei Atomicity beschrieben ist, ist das System in einem inkonsistenten Zustand und muss gegebenenfalls über Kompensationen repariert werden.

Isolation:

Durch die Reservierung der Ressourcen können andere Benutzer keine eigenen Reservierungen durchführen, die Transaktion ist dadurch bis zum Timeout isoliert. Es besteht jedoch die Möglichkeit, dass ein anderer Client die eigene Reservierung bestätigt und dadurch ungewollt eingreift.

Durability:

Falls die Transaktion bestätigt (confirmed) wurde, ist die Ressource und die Änderungen an ihr dauerhaft persistiert.